

IN THE CLAIMS

Please make the following amendments to the claims:

1. (Currently Amended) A method comprising:
maintaining a log of a plurality of requests in a storage server, each of the requests corresponding to a ~~storage~~write operation to be performed by the storage server on a set of storage devices, the log including a separate log entry for each of the requests; and
including a separate checksum in each of the log entries, each checksum for use by a checksum algorithm in determining data integrity of the corresponding log entry.
2. (Original) A method as recited in claim 1, wherein the requests originate from a set of client devices serviced by the storage server.
3. (Original) A method as recited in claim 1, further comprising selecting the checksum algorithm based on a desired balance between performance and checksum strength.
4. (Original) A method as recited in claim 1, further comprising automatically selecting the checksum algorithm based on a predetermined criterion.
5. (Original) A method as recited in claim 4, further comprising including an algorithm variable in the log to select the checksum algorithm from a plurality of selectable checksum algorithms, wherein said automatically selecting the

checksum algorithm comprises selecting the checksum algorithm dynamically by modifying the algorithm variable during operation of the storage server.

6. (Original) A method as recited in claim 1, further comprising:
 - including an algorithm variable in the log to select the checksum algorithm from a plurality of selectable checksum algorithms; and
 - automatically selecting the checksum algorithm dynamically by modifying the algorithm variable during operation of the storage server.
7. (Original) A method as recited in claim 1, further comprising including a separate algorithm variable in each of the log entries, to specify a checksum algorithm to be used separately for each said log entry.
8. (Original) A method as recited in claim 1, further comprising:
 - maintaining an entry count in the log to indicate the number of log entries in the log; and
 - using the checksum of one of the log entries to determine whether the entry count is corrupted.
9. (Original) A method of determining data integrity in a storage appliance coupled to a set of client devices and to a set of mass storage devices in a network environment, the method comprising:
 - receiving write requests from the client devices, the write requests requesting that the storage appliance write data to the set of mass storage devices;
 - saving data associated with the write requests to the set of mass storage devices at a consistency point;

maintaining a log of write requests received from the client devices since the consistency point in a non-volatile memory in the storage appliance, the log including a plurality of log entries, each of the log entries representing a separate write request;

including a separate checksum value in each of the log entries, each said checksum value for use by a checksum algorithm in verifying the data integrity of the corresponding log entry; and

including an algorithm variable in the log to select the checksum algorithm from a plurality of selectable checksum algorithms.

10. (Original) A method as recited in claim 9, further comprising selecting the checksum algorithm based on a desired balance between performance and checksum strength.

11. (Original) A method as recited in claim 9, further comprising automatically selecting the checksum algorithm based on a predetermined criterion.

12. (Original) A method as recited in claim 11, wherein said automatically selecting the checksum algorithm comprises selecting the checksum algorithm dynamically by modifying the algorithm variable during operation of the storage appliance in the network environment.

13. (Original) A method as recited in claim 9, further comprising automatically selecting the checksum algorithm dynamically by modifying the algorithm variable during operation of the storage appliance in the network environment.

14. (Original) A method as recited in claim 9, further comprising including a separate algorithm variable in each of the log entries, to specify separately a checksum algorithm to be used for each said log entry.
15. (Original) A method as recited in claim 9, further comprising:
maintaining an entry count in the log to indicate the number of log entries in the log; and
using the checksum of one of the log entries to determine whether the entry count is corrupted.
16. (Original) A network storage appliance comprising:
a first communication interface to receive requests from a set of client devices, the requests requesting that the network storage appliance perform storage operations in relation to a set of mass storage devices;
a second communication interface to couple the network storage appliance to the set of mass storage devices;
a non-volatile memory; and
a processor configured to cause the network storage appliance
to save data modified by the requests to the set of mass storage devices;
to maintain in the non-volatile memory a log of requests for which corresponding data has not yet been saved to the set of mass storage devices, the log including a plurality of log entries, each of the log entries representing a separate request, and
to include a separate checksum value in each of the log entries, each checksum value for use by a checksum algorithm in verifying the data integrity of the corresponding log entry.

17. (Original) A network storage appliance as recited in claim 16, wherein the processor is further configured to cause the network storage appliance to include an algorithm variable in the log, the algorithm variable indicating the checksum algorithm.
18. (Original) A network storage appliance as recited in claim 17, wherein the processor is further configured to cause the network storage appliance to set the algorithm variable based on a desired balance between performance and checksum strength.
19. (Original) A network storage appliance as recited in claim 16, wherein the processor is further configured to cause the network storage appliance to automatically select the checksum algorithm based on a predetermined criterion.
20. (Original) A network storage appliance as recited in claim 19, wherein the checksum algorithm is selected dynamically by modifying the algorithm variable during operation of the storage appliance in a network environment.
21. (Original) A network storage appliance as recited in claim 16, wherein the checksum algorithm is selected dynamically by modifying the algorithm variable during operation of the storage appliance in a network environment.
22. (Original) A network storage appliance as recited in claim 16, wherein the processor is further configured to cause the network storage appliance to include a separate algorithm variable in each of the log entries, to indicate a checksum algorithm to be used separately for each said log entry.

23. (Original) A network storage appliance as recited in claim 16, wherein the processor is further configured to cause the network storage appliance to maintain an entry count in the log to indicate the number of log entries in the log, and to use the checksum of one of the log entries to determine whether the entry count is corrupted.

24. (Currently Amended) A storage server comprising:
means for receiving a plurality of requests from a set of client devices; each request corresponding to a ~~storage~~write operation to be performed by the storage server in relation to a set of storage devices; and
means for maintaining a log of the requests in the storage server, the log including a separate log entry for each of the requests, the log further including a separate checksum in each of the log entries, each checksum for use by a checksum algorithm in determining data integrity of the corresponding log entry.

25. (Original) A storage server as recited in claim 24, further comprising means for selecting the checksum algorithm based on a desired balance between performance and checksum strength.

26. (Original) A storage server as recited in claim 24, further comprising means for automatically selecting the checksum algorithm based on a predetermined criterion.

27. (Original) A storage server as recited in claim 26, further comprising means for including an algorithm variable in the log to select the checksum algorithm from a plurality of selectable checksum algorithms, wherein said means for automatically selecting the checksum algorithm comprises means for

selecting the checksum algorithm dynamically by modifying the algorithm variable during operation of the storage server.

28. (Original) A storage server as recited in claim 24, further comprising:
an algorithm variable in the log to select the checksum algorithm from a plurality of selectable checksum algorithms; and
means for automatically selecting the checksum algorithm dynamically by modifying the algorithm variable during operation of the storage server.
29. (Original) A storage server as recited in claim 24, further comprising means for including a separate algorithm variable in each of the log entries, to specify a checksum algorithm to be used separately for each said log entry.
30. (Original) A storage server as recited in claim 24, further comprising:
means for maintaining an entry count in the log to indicate the number of log entries in the log; and
means for using the checksum of one of the log entries to determine whether the entry count is corrupted.
31. (Original) A storage server as recited in claim 24, wherein the storage appliance is a network appliance.